UCID- 21234

THE CONFLICT MODEL (CONMOD)
INTERIM TECHNICAL OVERVIEW

Yu-Hsing Chiu
Ronald L. Burback
Charles D. Jones
Robert L. Pancotti
S. Christian Simonson
Sara M. Strange
Ralph M. Toms

October 1987

Lawrence Livermore National Laboratory

This is an informal report intended primarily for internal or limited external distribution. The opinions and conclusions stated are those of the author and may or may not be those of the Laboratory.
Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

DISCLAIMER

Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

Price Code	Page Range
A01	Microfiche
Papercopy Prices	
A02	001-050
A03	051-100
A04	101-200
A05	201-300
A06	301-400
A07	401-500
A08	501-600
A09	601

Table of Contents

Appendices

Figures

The Conflict Model (ConMod)
Interim Technical Overview

1.0 Executive Summary

The Conflict Model (ConMod) is an automated, large-scale, air/land conflict simulation at the corps and echelons-above-corps levels. Currently in development, the first release is scheduled for Fall 1988.

ConMod is an analytic computer model for addressing operational concepts of air and ground forces engaged in deep, rear, and close combat. While designed as an analytic tool, ConMod is also adaptable to research, operational support, and training purposes.

The model simulates selected aspects of combat, combat support, and combat service support in an air/land battle. It covers a geographical extent of hundreds of kilometers, with a time period of days.

Key concepts of the model include cause and effect audit trail, discrete event simulation with stochastic physics processes, separation of cognitive and physical activities, interruptible execution, and extensible architecture. Command and control is an explicit and integral part of the simulation.

ConMod takes an object oriented approach to the modeling of military systems, founded on control systems theory. Generic military objects with scenario dependent links form military organization hierarchies and C^2 networks. Planning may be automated below a selectable organizational level.

ConMod uses high resolution physics for the simulation of movement, acquisition, engagement, and communication. Physics calculations are based on item systems (i.e., major pieces of equipment). Engineering data is used for these calculations. ConMod uses 3D digitized terrain, typically on a 100-meter grid.

The model has a comprehensive system support environment. ConMod is written in the Ada® programming language. ConMod is targeted at computer systems of the VAX 8000 class with 48 MB central memory, and its high-resolution, color graphics user interface runs on Tektronix 4120 series workstations.

2.0 Introduction

The Conflict Model (ConMod) is an automated, high-resolution, echelons-above-corps (EAC) conflict simulation, which is in development at the Conflict Simulation Center (CSC) of Lawrence Livermore National Laboratory (LLNL). The project is jointly sponsored by the Army and LLNL, with Air Force participation in the form of two officers assigned to the project team.

This paper presents an overview of ConMod in general technical terms, at an interim stage of its development. It is based primarily on presentations made to the Army-LLNL Technical Steering Committee meeting on July 16, 1987, and on some material from previous briefings and documentation. At that time the basic concepts had been fixed, the model had reached the design stage, and implementation had started in several areas. It is recognized that the details in a number of areas will be further worked out as development progresses. Hence this is referred to as an "interim" technical overview.

This paper is intended to provide prospective users of ConMod with an understanding of its basic characteristics. By "prospective users" three groups are included: those who will work with ConMod directly; those who will interact with the model, such as in operational support or training modes; and those who will use the results of studies made with ConMod.

In the remainder of this introductory section, the project origins will be discussed, together with development plans, and potential applications. In the following section, the requirements expressed by the Army and the Laboratory will be summarized. Next, the design concepts of ConMod will be described. In the rest of the paper, these will be elaborated in two major areas, software design and simulation design.

In the software area, first the software development methods used by the project will be discussed. Then the system support environment will be described. In the simulation area, first the model architecture will be described. Then the major new modeling area, command and control (C^2) simulation will be discussed, including the automated aspects of C^2 and its application at the operational level.

Input and output data requirements are then discussed, and an estimate of the required computer resources concludes the paper.

2.1 Background and Development Plans

ConMod originated in response to a need for high resolution combat modeling coupled with large scope. High resolution is needed for the evaluation of individual weapons systems. Large scope is needed to represent the physical environment, the time scale, and the C^2 structure in which the weapons systems operate.

By the mid-1980's it was thought that computer development had reached the point where these goals were achievable on widely available minicomputer systems. The U.S. Army had needs similar to those of LLNL and joined in financially sponsoring the project as well as in assigning three officers to the CSC. A Memorandum of Agreement was signed in August 1985 between the Army, the Laboratory, and the Department of Energy (Appendix A). The U.S. Air Force also assigned two officers. An initial three year development began in 1985, which was planned to result in the release of a version capable of limited studies in late 1988.

ConMod is being developed in phases. This applies both to the purely software aspects of the project and to the military functional areas. Phases in the incremental development include concept definition, prototype development, and production release. The analysis of requirements and the software design were completed and documented (Ref. 6-11) prior to code generation.

An internal prototype will be constructed during calendar year 1987 with a limited set of functions to prove certain critical concepts. It will be augmented and extended to become the first release.

In the military area, methods have been put in place to capture military data in functional areas. The first areas to be developed will be close combat, air-to-ground, air defense, fire support, and intelligence/electronic warfare. Later will come engineer, logistics, etc. These are being developed in a joint and balanced approach. For example, air and air defense will be represented equivalently. Furthermore, both Red and Blue sides will be treated equivalently, while maintaining the distinguishing asymmetries between them.

The first release in October 1988 will have limited functionality. Although capable of limited studies, its purpose is to provide an opportunity for users to become familiar with it and to initiate a testing program. An optimal plan would be to have it in a single location as a beta test site for a lengthy period during the first year.

A version capable of full corps-level studies will require further effort extending into 1989 and beyond. While the testing of the first release is in progress, additional functions can be developed for this more general second release. It would also incorporate the results of the testing program. Further releases can extend the number and depth of functional areas and incorporate additional user requirements.

2.2 ConMod Applications

ConMod is designed as an analytic tool to enable the user to examine activities of a military conflict involving air and ground forces. It is primarily intended for evaluating existing and proposed combat systems, tactics, and doctrine in order to provide timely evaluation and recommendations to policy makers.

Because of its interactive and interruptible features, ConMod has an inherent potential for adapting to the needs of research, operational support, and training purposes as well as analysis. At the highest level, human-generated plans drive the simulation. It will also be possible to interact with the simulation during its execution by manually taking over an automated planner's role. This allows the control of major decisions of particular interest to a study. The model is designed to be interrupted at designated conditions or upon user request and the state variables saved. It may then be restarted with new planning input so that excursions from key events can be examined.

As an example, in the training mode human players can take over a subset of the cognitive tasks. For instance, players might assume the logistics planning role. The other tasks would be automatically planned and executed. The logistics players can then see the effects of their actions in the simulation without tying up staff resources playing the numerous other roles. This mode would also allow a certain degree of validation of the various functional areas by running ConMod with predefined plans and also with a group of expert players and then comparing the results.

Using a single basic model over these major areas offers several advantages. A primary one is consistency of high quality algorithms and data among the various users. This helps to ensure that lessons learned and insights gained in one area of application remain valid when looking at another area. Furthermore, the research, training, and operational support users would benefit from the analytical model's assessment capabilities.

3.0 General Requirements

Both the Army and the Laboratory have expressed sets of general requirements for ConMod. Requirements were described in broad terms in the MOA (Ref. 1, see Appendix A). The Army proposed an overall set of requirements, which after discussion with the Laboratory were issued in revised form as Ref. 2. Later, the Laboratory enunciated its own particular requirements (Ref. 12). Pertinent extracts from these documents are given here. Based on these guidelines, the ConMod project team developed specific, detailed requirements, which are documented in Ref. 6-8.

3.1 Army Requirements

The following extracts from Ref. 2 state the general Army requirements.

"This document provides the Army Requirements for the Joint AirLand Corps combat model to be developed by Lawrence Livermore National Laboratory (LLNL) with co-sponsorship by the Army as described in the Memorandum of Agreement (MOA) between the Department of the Army and the Department of Energy and LLNL. As a general requirement, the design, development and implementation will have a structured design, and will be compatible with the INGRES database management system [Ref. 14]. Where departures from these requirements are identified in the course of design or model development they will be documented and approved by the Technical Steering Group.

"The Army looks to this model to provide analysis support capabilities well beyond those offered by stabilized versions of existing corps level models. This emphasizes the intention to exploit advanced/ state-of-the-art hardware and software capabilities in regard to software maintenance, database management, computing capacity, and ease of analyst interface with highly visible model results from simulated battles. Generic capabilities to be considered are detailed in the requirements definitions which follow.

"The Army requirements provided here are of three types – functional representations which will support the analysis issues the model must answer, technical performance specifications, and model development processes. This document is the Army's set of model requirements for the joint Army-LLNL effort. It is intended to become a Joint Army-LLNL set of requirements that will be used to guide the model development effort. . . .

"The Army-LLNL model will portray the ground and air combat, combat support and combat service support elements expected to be present in a friendly or allied corps or division and opposing Threat forces at the Army level in the 1988-2000 time frame. The model will have the capability to assess force effectiveness as a function of changes in doctrine and tactics, equipment, organizations, system performance, or degraded human performance conditions (suppression, fatigue, training level). The effects of factors external to the modeled forces which impact on the friendly and enemy forces, such as national asset intelligence information and the allocation and apportionment of air support, will be included in the model. The model will consider the interaction of the simulated forces with the environment (terrain, vegetation, obscurants, weather, day/night) and allow for variations in these factors. The deep and rear battle and support functions will be represented. Each functional area will interact with other appropriate functional areas. . . ."

[The Army required representation of the following functional areas:

Command and Control
Intelligence
Communications
Electronic Warfare
Close Combat
Fire Support
Aviation
Air Defense
Engineer
Logistics/Personnel Support
Nuclear/Chemical Combat
Environment

In each of these areas, Ref. 2 provided a discussion of the critical processes and functions, the required entity resolution and interactions with other functional areas, and gave the rationale with examples of analysis issues.]

". . .In order to provide the computing capacity, speed of execution, software maintenance, ease of analyst/user interface and high resolution to the element or small unit level, the model will exploit, to the extent possible, advanced and/or state-of-the-art hardware and software technologies. Structured design approaches will be utilized for software system design/implementation. Stochastic representations of specified factors known to produce battle outcome variability in the real world should be available. Techniques should be used to ease the analyst/user interface with the model. The model will support advanced graphics capabilities and use a relational database management system (INGRES). Data requirements for the model will be provided through available Army sources; new data requirements, unique to the model, will be described by a detailed description of how to generate such data. . . .

"The Army's continued interaction with LLNL throughout the design, development and implementation phases of this effort [will be ensured.] . . . Documentation is to be prepared throughout model development . . ."

3.2 LLNL Requirements

The following delineates the minimum requirements of the Laboratory, together with the general rationale behind them (extract from Ref. 12).

"[Two specific groups at LLNL] have requirements for corps and echelons-above-corps (EAC) simulations. . . . The Theater Applications Group conducts analytic assessments of the effectiveness of alternative nuclear and conventional theater weapon systems. . . . To make these assessments, the Theater

Applications Group requires models that (among other things): (1) capture the details of the technology under consideration at the subsystem level, (2) provide the capability to perform rapid parameter analyses, (3) permit specification of all relevant system and subsystem parameters, (4) permit the specification and control of scenarios and target arrays, and (5) provide statistically significant and reproducible results. . . .

"The Conflict Simulation Center provides an environment for examining the utility of current and future tactical nuclear weapons, force configurations, and deployment issues. . . . The key requirements for the analysis of these issues include the following:

"Representation of Weapons.--The Laboratory's primary concern is for nuclear weapons modeling capability in the battlefield environment. The Laboratory also needs to be able to model improved and advanced conventional munitions (ICM and ACM), both by themselves and in connection with nuclear weapons. In addition, the Laboratory must evaluate advanced physics concepts, such as directed energy weapons. Mixes of weapons systems need to be considered, including complementary as well as supplementary or replacement systems.

"Scope: Corps and EAC Level.--The effects of nuclear weapons cover such a large area and affect such a variety of systems that corps level modeling is needed to establish the appropriate environment even for tactical weapons. The control of nuclear weapons and their delivery systems is maintained at high echelons and involves both Army and Air Force, thus requiring the modeling of EAC command and control structures.

"Time Period and Geographical Scale.--The simulated engagement period must be long enough and the scale large enough to see follow-on forces enter the battle. This means covering an extent of 200-400 km from the forward line of own troops and spanning a duration of a minimum of 2-3 days. The examination of intermediate dose effects requires somewhat longer times. Survivability issues require similar representation of the rear area.

"Representational Detail.--The ability to fix cause and effect is critical to the utility of any simulation as an analytic tool. This implies an appropriate resolution of events in both space and time. The model must be capable of examining the effects of stochastic processes. Individual weapons systems need to be modeled to conduct tradeoffs. Thus selective item-system resolution is implied. Since the goal is the evaluation of weapon effectiveness, effectiveness must appear as an output, not an input. The weapon system input must be restricted to engineering data, as distinguished from effectiveness data. The capability to readily change the data is required to efficiently perform sensitivity studies. When line-of-sight or terrain masking aspects of a problem are important, a three-dimensional terrain is required in the simulation. The terrain model is also important in accounting for the locations of discrete events when assessing cause and effect.

"Model Employment Considerations.--The Laboratory has limited resources to call on for military expertise. Consequently there is a need for automated command and control with the ability to accept plans prepared by others ahead of time. The ability to mix and match scenarios is essential to allow alternative offensive concepts to be played against alternative defensive concepts.

"System Requirements.--To make the best use of limited analyst resources, the input data for the model must be easy to set up and modify, and the output must be capable of rapid analysis and presentation. This implies the use of interactive color graphics systems to provide user-friendly interfaces. To minimize the costs of development and maintenance, good software engineering and development practices are required. The code must be easily extendable and transportable to alternative computer systems. For ease of access during development and employment, the simulator should be capable of running on a dedicated, stand-alone minicomputer.

"Model Integrity.--The Laboratory is concerned with assuring the consistency and reproducibility of results so that study results can be compared and assessed against each other. The code and the data must be documented and kept under configuration control. Tests to verify that the code meets its specifications must be run and documented. The goal is to validate the model and improve its algorithms by comparison with experiments and field data."

4.0 ConMod Design Concepts

ConMod emphasizes five significant features in its conceptual design. These are: (1) C^2 representation based on control system theory, (2) separation of the cognitive aspects of the simulation from the physical aspects, (3) cause and effect audit trail, (4) discrete event simulation, and (5) extensible model architecture. The rationale for these features stems from ConMod's objectives. Each of these will be discussed in turn, bearing in mind that the aim of this or any simulation is to represent those characteristics of the system that are pertinent to the problem under study.

4.1 Theoretical Background

For automated command and control, a control theory approach to organizations provides a theoretical foundation. In this structure, the cognitive aspects of the problem, namely, C^2, can be viewed separately from the physical aspects. The control system approach also facilitates looking at issues of stability. Within this overall organizational framework, the control agent itself is modeled using a theory of management.

4.2 Cognitive Plane and Physical Plane Separation

The organizations interact on two separate but related planes: the cognitive plane and the physical plane. Each independent organization has its own cognitive plane, but it shares a physical plane with all other independent organizations. Thus there is a single physical plane but a cognitive plane for each of the opposing forces.

This separation of the problem recognizes two distinct types of modeling effort. The physical plane deals with physical processes such as sensing, moving, engaging, and communicating. The cognitive plane emphasizes management processes such as directing, controlling, and coordinating organizations.

4.3 Cause and Effect Audit Trail

In a model whose purpose is analysis, the ability to identify cause and effect is vital. In the ConMod design, a mechanistic viewpoint is imposed whereby all effects have a known cause and all effects are calculable. This is achieved by requiring two entities: objects and actions. When two objects interact through an action, there are also two events: the cause event and the effect event.

4.4 Discrete Event Simulation

The need for a cause and effect audit trail combined with the need to examine individual item systems leads to a discrete event simulation. ConMod is conceived as an event driven, variable resolution model. The simulation proceeds through the execution of scheduled (queued) events. One event is either an object initiating an action (cause event) or an object being acted upon (effect event).

Since ConMod resolves down to selected item systems on digitized 3-D terrain, it becomes possible to use actual locations for determining range and range-dependent variables, such as the probability of hit and probability of kill (P_h/P_k). This allows cause and effect to be established using the actual locations and actual times for discrete events, particularly sensing and engaging. These low-level events are modeled stochastically.

4.5 Extensible Model Architecture

Constant change is the norm of the military world. In order to accommodate the future changes in weapons systems, organizations, operations, tactics, and doctrine, ConMod has adopted an object oriented development method. Since a clear distinction is made between cognitive and physical processes, future extensions that utilize knowledge based system concepts can be facilitated.

5.0 Software Development Methods

The software objectives of the ConMod development include the use of modern software development practices and state-of-the-art computer science and programming techniques. These are meant to minimize the costs of the overall software life cycle, particularly in the maintenance phase (including enhancements and extensions).

ConMod's development addresses the "software crisis" through the use of the following methods:

Extensive analysis and design phases
Documentation
Highly structured language (Ada)
Object oriented methodology
Quality assurance program
Testing program for verification
Configuration control for consistency

In the analysis and design areas, joint meetings are being held among the intelligence community, the Laboratory community, and the military community to define requirements. Joint analysis and design reviews have also been held.

The ConMod project has published documents as the model has been developed. These include requirements documents (Ref. 2 and 6-8) and design documents (Ref. 9-11), as well as a software configuration management plan (Ref. 4) and a software quality assurance plan (Ref. 5).

The Ada programming language was chosen for the ConMod project because it encourages good software engineering practices. It supports highly structured code and lends itself to the extensibility objective.

Ada was adopted after careful analysis (Ref. 15). It won out over alternatives because it supports modern practices. A feature that is important in large software projects is the separation of externally visible specifications and internal processes among program modules. The internal body, including algorithms and private data, can be changed without changing the specification. Thus a change doesn't necessarily propagate anywhere else in the model. This also allows parallel efforts to proceed independently once the interfaces are defined. Another strong point of Ada is the support of reusable, generic code. Finally, use of a standard language allows transporting the code to alternative computer systems, and Ada is the Department of Defense standard.

The object oriented methodology was selected to foster cohesive, loosely coupled software modules related to understandable, recognizable objects. It was also chosen in order to support the extensibility objective and the maintainability goals.

In object oriented design, data and algorithms are encapsulated in a program object. The advantage of object oriented design as opposed to process oriented design is that it resembles the real world. It is easy to modify, to add or delete. A corollary is extensibility. A small portion can be built first, then extended. Also, code designed this way is highly reusable.

The last three items, quality assurance, testing, and configuration control, have to do with enforcing the goals. The quality assurance program, based on the published quality assurance document, is governing the team's work. Periodic walkthroughs and inspections are held on analysis and design documents as well as on code modules. A testing program for verifying the correctness of the code is planned. There is a vital interest in configuration control to maintain consistency, and it has been established from the beginning rather than as an add-on at the end. Ada is expected to help with configuration management.

LLNL is striving to create a development environment conducive to success. The Laboratory is supporting the ConMod project in several important ways. One is the purchase of a dedicated DEC VAX 8800 computer, together with the DEC Ada programming environment and other software, for the Conflict Simulation Center. The VAX 8800 was chosen as the development machine for several reasons, among them being the installation of the related VAX 8600 computers at the unified and specified commands as part of the Modern Aids to Planning Program of the Organization of the Joint Chiefs of Staff. The model is designed to be able to run on the VAX 8600, although with possibly decreased performance. The VAX 8800 is designed to support parallel processing, with dual central processing units.

The Laboratory has assembled a team of qualified personnel for the project, and has invested in training and in equipment, such as terminals, workstations, and automated software engineering tools. It has also provided consultants, both from within the Laboratory and from outside. These represent approaches to reducing the risk of moving in new directions in software development.

6.0 System Support Environment

The system support environment (SSE) facilitates simulation setup, execution, and analysis. Run-time interaction with an executing simulation permits the augmentation of certain setup efforts and the trending of intermediate results. The system support environment manages all computer resources needed to support study activities.

6.1 Setup, Execution, and Analysis

Access to both default and previous study setup information minimizes the time dedicated to setup. Execution of a simulation can be started from a branchpoint, e.g., a critical decision point, established in a previous simulation. Execution can be terminated or suspended manually by the analyst; similarly, the analyst can specify conditions which can cause this termination or suspension automatically. A selectable history is archived automatically. Optionally, reports can be generated during run-time and at post-run. A set of utilities will be provided to the user for setup, monitoring the execution, and performing post-run analysis.

6.2 Interactive Color Graphics Interfaces

The support environment supports two distinct human interfaces and their associated activities, corresponding to the analyst and the planner roles. Simple-to-use user interfaces for both the analyst and the planner are key elements of the ConMod system. High resolution color graphics workstations with menus and icons are the main ingredients of the user interfaces.

6.3 System Support Environment Description

The design objectives of the SSE are the following:

Maintain independence of simulation and support activities.
Perform support activities in parallel with the simulation.
Optimize movement of data through the system.
Provide for flexible development and evolution of supervisory states and user interfaces

In comparative terms, the SSE may be described as: asynchronous (vs synchronous), multiple (vs single simulation knowledge), operational (vs computational), supervisory (vs process), service and data collection (vs data generation).

The simulation can initiate requests through SSE components and is independent of underlying database technology. The users interact with the simulation through a set of utilities. The utilities define a set of requests that can be made, which typically result in some kind of information transfer (for use in generating reports, displays, etc.). The requests are directed to objects in the simulation, which respond by making data transfers to the SSE. Data transfer (history files, etc.) will be transparent to the user. The SSE preserves independence of activities. Support activities work in the operational domain, responding to conditions in the simulation and to user interaction.

Initialization is in some sense a converse of a user data request. Rather than transferring information out of the simulation, the simulation requests specific kinds of data from the database, which transfers into the simulation domain.

The SSE capabilities can be categorized as constituting the following:

User interaction.
Data processing, presentation, and storage.
Data collection, initialization, and termination.
Control of batch operations.
System state management.
Database administration.

The control subsystem manages (or supervises) both the support environment itself and the simulation. System state can be changed by analyst request or in response to simulation conditions. SSE functionality includes batch operation, gaming operation, initialization of the simulation prior to execution, extension of plans and scripts during suspend, analysis during runtime and post runtime.

Runtime controls include selectable reports that are automatically generated during runtime, interruptability of the simulation (including suspend, possibly for an indefinite peiod of time).

User interfaces are broken down into functional components that are activated or deactivated according to system states. For example, these include runtime analysis, post run analysis, and different kinds of setup or plan development at pre run. A host of utility displays, such as ground truth, etc., are supported.

7.0 Model Architecture

ConMod's model architecture emphasizes flexibility and extensibility. It is designed to accommodate changing requirements. It encourages phased evolutionary development, including addition of military objects and features, and the addition and improvement of algorithms. It also supports prototyping.

Changing requirements are accommodated by allowing the addition and integration of new objects. The use of Ada facilitates changes to existing objects. Ada libraries automatically identify the coupling between revised objects and other objects in the software.

The result of this is that it encourages evolutionary development. This is the only practical way to develop this type of simulation because of the complexity and scope of the problem. Evolutionary development can be done by adding new objects and adding new features to existing objects.

Each new object or feature added to the program can be added in a prototype form and can thus be gradually integrated. This allows the model to have a long lifetime, because it can be constantly growing in its features.

7.1 Simulation Model Design

The simulation model makes use of object-oriented development, particularly because it provides a methodology for effectively addressing data abstraction and information hiding.

The architecture of the system is organized around the concept of objects and classes of objects that parallel our model of reality. An object is an entity that has state, is characterized by the operations that it absorbs and initiates, and is an instance of a class of objects.

Object oriented techniques map well into Ada. In particular, the Ada package supports the object oriented approach. Ada also has a feature known as a task, which operates in parallel with other parts of the program. In ConMod, the use of tasks will be allowed liberally in the System Support Environment (SSE), but prohibited in the simulation, except for what is necessary to interface with system support environment tasks. For repeatability, within the simulation itself events must happen in the same order. The use of multiple tasks and therefore concurrent operations does not necessarily guarantee this.

A simulation object can be a generic military object or an instance of such. An example of a generic military object is a ground mission authority (GMA). It interacts with cognitive terrain and does direction, planning, reporting, coordinating, and communicating.

Cognitive ojects have only one physical capability, communicating. On the other hand, a physical object, for example, a close combat unit, has much less internal processing but has more physical processing. Objects commmunicate among each other, directing and reporting along hierarchical organizations, and coordinating across hierarchies.

The C^2 objects in the cognitive plane have force system counterparts in the physical plane, namely, command posts. Command posts perform the physical process of communication. They may be engaged by combat systems and damaged or destroyed, and this will affect the performance of the C^2 function.

7.2 Physics Modeling Characteristics

Plans of a C^2 object are converted into explicit directives and are issued to its subordinates. These in turn issue lower level directives to their subordinates until this reaches down to force system objects. Then they are carried out by a set of physical actions.

The physical actions in ConMod are modeled using item-system resolution on three-dimensional digitized terrain. The weapons system input is restricted to engineering data, as distinguished from effectiveness data. This includes characteristic parameters for moving, sensing, engaging, and communicating.

By resolving down to selected item systems on digitized 3-D terrain, ConMod will be able to use actual locations for determining range and range-dependent variables, such as the probability of hit and probability of kill. In support of one of the key goals, cause and effect can be established using the actual locations and actual times for discrete events, particularly sensing and engaging.

8.0 Command and Control Simulation Characteristics

In this section, the important simulation features relating to command and control will be further developed. First the control system and organizational theory are applied to combat simulation. Then the cognitive/physical separation is described. Next military organizations are shown as they fit into the structure, and their management functions are described. (Additional details are given in Ref. 13.)

8.1 Control System Theory

A system can be defined as a group of objects interacting with each other through well defined actions and behaving as a unified whole with respect to the system's environment.

A control system is composed of two subsystems: (1) a controller and (2) a producer. The controller attempts to control the producer's behavior in the presence of environmental interactions.

A metacontrol system is a special kind of control system. Metacontrol is the control of a controller. This has the effect of distributing control through various levels, as is commonly done in organizations.

An organization behaves as a control system. It attempts to control its producers in the presence of interactions with the physical environment. An organization is the union of a management metacontrol system and a production control system.

A hierarchical organization has a layering of management metacontrollers in its management metacontrol system in order to provide the desired span of control of a number of specialized production control systems. Typical organizational structures may be constructed by combining features from a centralized structure and a decentralized structure. In the centralized organization, high level managers may exert control down several levels, including control of production controllers, while in the decentralized organization, high level management controllers only control other management controllers.

8.2 Cognitive and Physical Separation

As described previously under design concepts, the C^2 system in ConMod is separated from the physical combat processes. In the model, this is expressed in terms of planes--one physical plane and two cognitive planes, one for each opposing side in the conflict. This is shown in Figure1. In terms of organizational theory, the C^2 system is the management metacontroller, and the force system is the production control system.

The C^2 system is composed of cognitive objects related by cognitive actions. It lies entirely in the cognitive plane of its respective side. The configuration of a C^2 system may be customized to reflect a particular hierarchical organizational structure.

The force system is composed of active and passive physical objects. Active physical objects include such things as single weapon systems, tactical groupings (aircraft flights, tank platoons, etc.), command posts, logistics centers, and communications centers. Passive physical objects include such things as unissued supplies, unissued equipment, unassigned personnel, and barriers. The force system may interact with the C^2 system through management actions such as directing, coordinating, and reporting. The force system also interacts with the common environment system and may cause physical actions which affect objects in the common environment.

Common environment objects lie in the physical plane. They include terrain, vegetation, hydrographic features, and cultural features, as well as lethality fields, weather fields, radiation fields, chemical contamination fields, etc.

It should be noted that the interaction between two opposing sides occurs only in the common physical plane. There is no direct connection between cognitive planes. Thus ConMod excludes what might be termed political processes, such as direct negotiations between the cognitive parts of opposing sides. Blue has no way of directly manipulating Red's cognitive processes. This implies, for example, that if Blue wants to deceive Red, it has to manipulate objects or events in the physical plane that Red might misinterpret.

8.3 Generic Objects in Military Organization Hierarchies

The extensibility objective of ConMod's development requires a conceptual architecture and software design which will allow for continuous expansion of the number and kinds of military functionalities represented. It is recognized that the partitioning of military activities into functional areas is largely doctrinal. Military entities typically perform tasks in more than one functional area. In consideration of differing organizational doctrine on both sides as well as to allow for future developments, the discussion of military entities will be couched in generic terms.

The object oriented approach satisfies these needs for flexibility and extensibility. This proceeds as follows. From an examination of the military entities of interest, a set of generic simulation objects sufficient for the problem is abstracted. Each generic object can be viewed as a finite state machine. It has a unique set of characteristic and state attributes. Each generic object is then further viewed as residing in a control system, receiving and sending signals which affect its state or alter the states of other generic objects. The signals become the generic events of a discrete event simulation.

Objects within the force system of each side are tactical groupings appropriate to the resolution of the simulation. Examples are artillery batteries, command posts, and aircraft flights. Active force system objects are capable of performing specialized tasks in the physical plane. One way to express this specialization is to consider that each force system object has its own specific language. For example, artillery batteries use a language that is distinct from that used by aircraft flights. The language specific activities performed in the C^2 system are represented by what may be termed authority centers.

The cognitive authority centers can be mapped into real world military objects. The mapping allows flexibility in designating, for any particular force structure, who performs a specific cognitive activity. By closely relating these objects to real world entities, such as command posts, their behavioral characteristics can be elucidated.

Two types of authority centers have been included in the ConMod concept: (1) control authorities at the lowest level, and (2) mission authorities at higher levels. These are shown in Figure 2.

A control authority exercises tactical control over a group of specialized force system objects. Examples are fire support control authorities, close combat control authorities, and air defense control authorities. Control authorities receive an operation order from a superior and attempt to execute the order by issuing detailed tasking to assigned force system objects. Control authorities report their status to their superior and may request support for their operation through coordination channels when authorized.

Mission authorities exercise operational control over subordinate forces. They receive a broad directive which includes allotments of forces and resources. The language of mission authorities reflects the types of operations their subordinate control authorities can execute. Mission authorities may also control other mission authorities of the same type. This is indicated in Figure 3. Examples are force mission authorities, ground mission authorities, and air mission authorities. Mission authorities issue mission directives and operation orders to their subordinates. They may request support for their mission through coordination channels when authorized.

8.4 Management Processes for C^2 Objects

Each C^2 object is conceived of as a management entity capable of performing five processes: (1) planning, (2) directing, (3) controlling, (4) reporting, and (5) coordinating. These are illustrated in Figure 4. Management processes must be customized for a particular C^2 object; however, the data flow between processes is generic to all C^2 objects.

Key to the management process is the local data maintained by each C^2 object: (1) the plan, (2) the perceived situation, and (3) policy data. The plan may be either the result of an automated planning process or, for those objects in a manual planning mode, a manually prepared plan. Plans contain constraints imposed by a superior on its subordinate through a directive. The perceived situation is updated from information received through feedback and coordination. The perceived situation has three aspects: (1) environmental perception, (2) threat perception, and (3) friendly perception. The environmental perception includes current knowledge of objects in the common environment. Threat perception includes current knowledge of objects in the opposing organization's force, fused to the appropriate level for planning. The friendly perception includes current knowledge of other objects in the same organization, including immediate subordinates. Policy data is characteristic data used by the management processes. Policy data contains information with doctrinal and procedural implications.

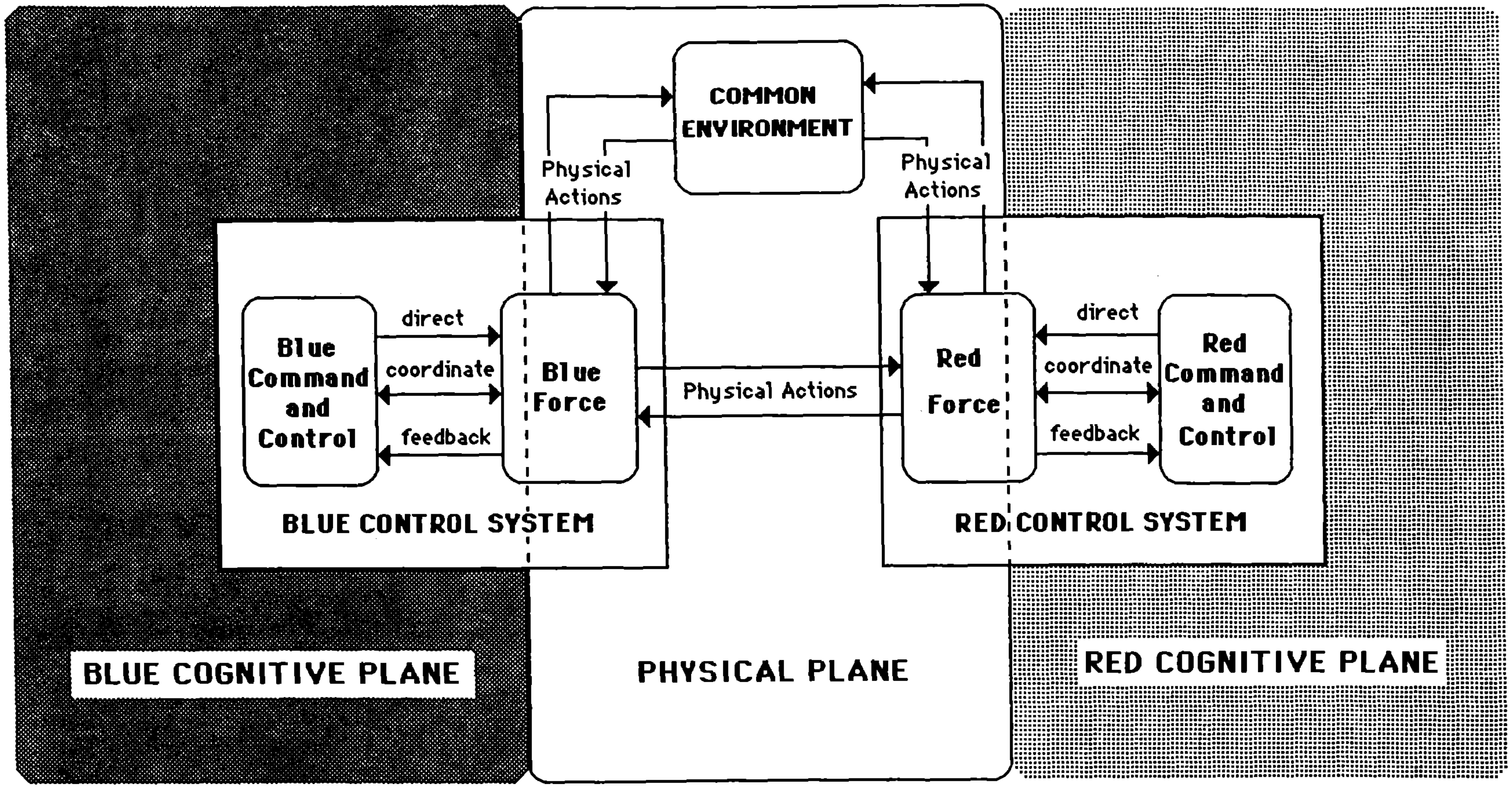

Figure 1. Competing military control systems, with separate cognitive planes and common physical plane

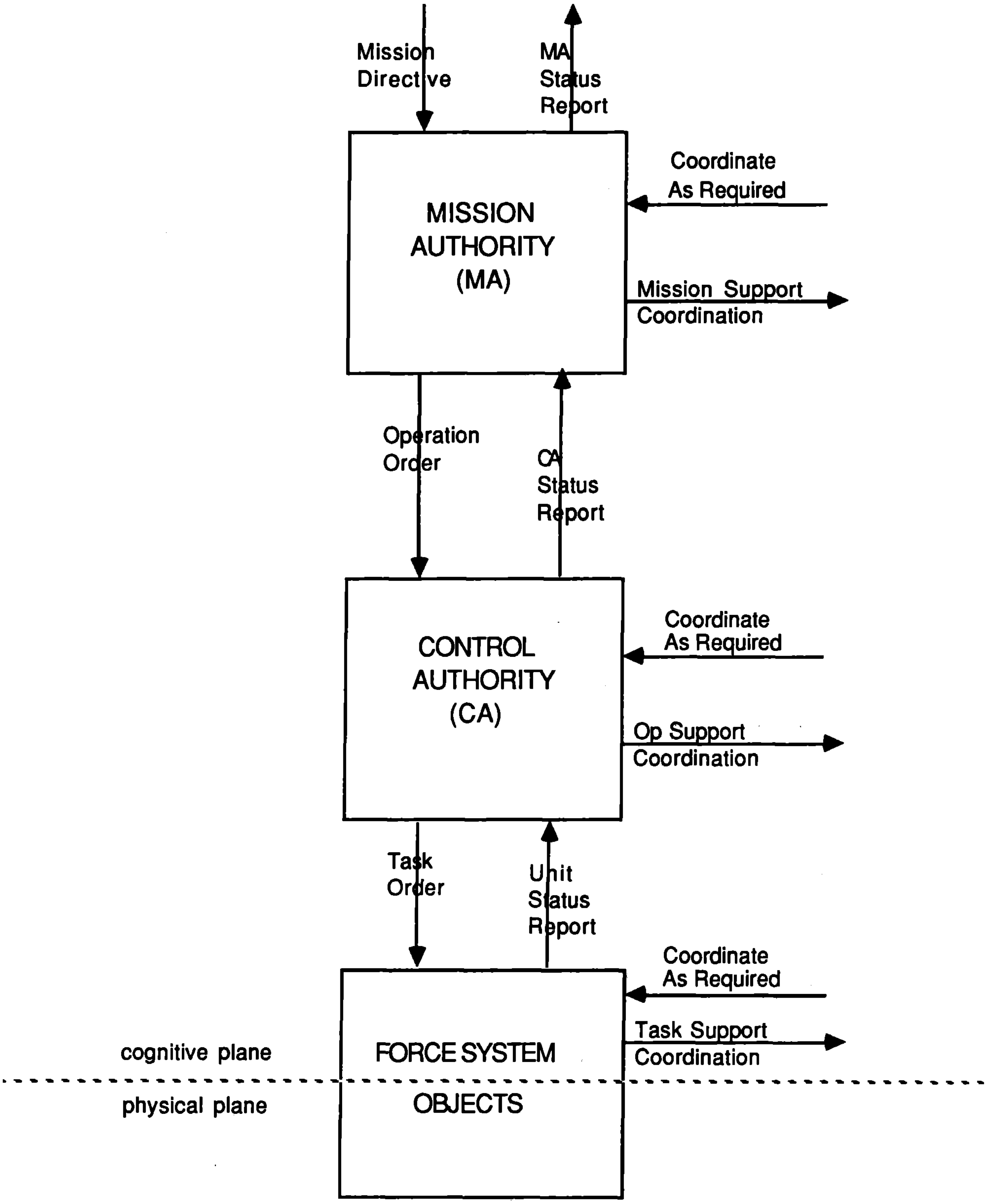

Figure 2. Military Management Functions

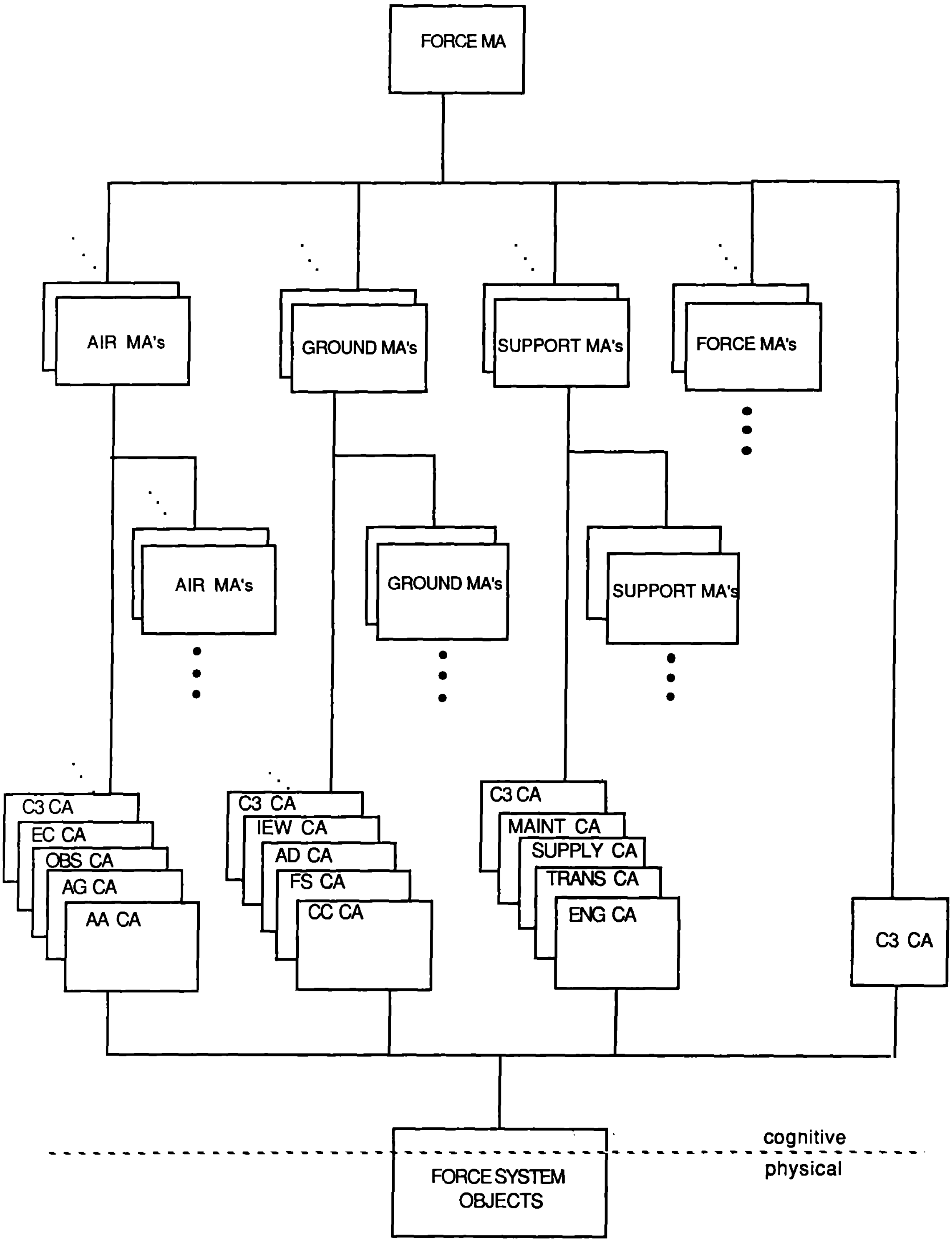

Figure 3. Generic Military Organizations

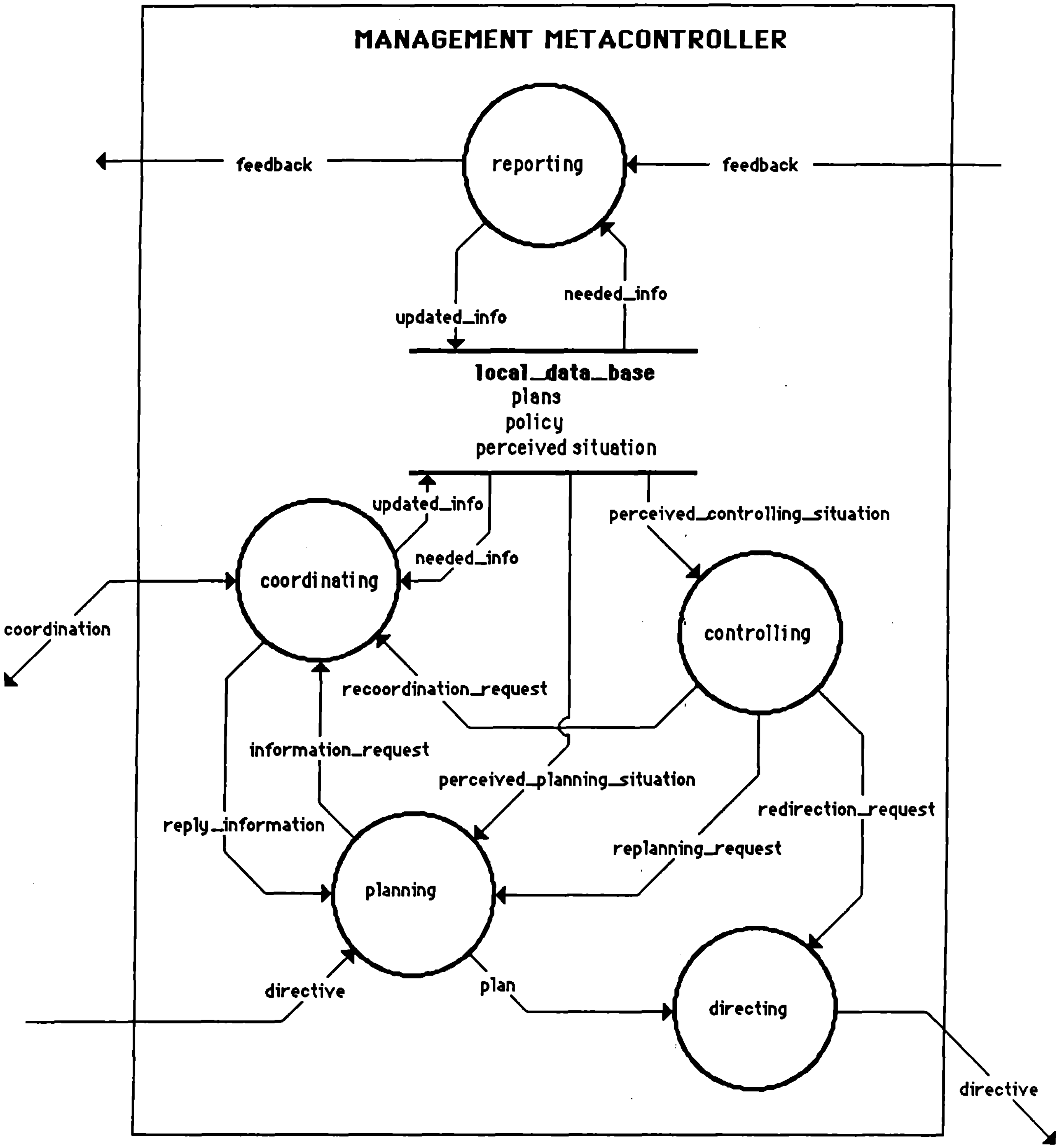

Figure 4. Process Model of Management

9.0 Manual and Automated Planning

9.1 Human Planner

ConMod is conceived as having two roles to be filled by the persons interacting with the simulation. The first role is that of the analyst, who controls the simulation. The second role is that of the planner, who prepares the military planning input.

The analyst sets up the simulation for a particular study. Planners assist the analyst by developing military plans and policies which support study objectives. The planning activity is constrained by the planner roles and the organizations designated by the analyst for use in the study. A planner has access to only that information which is relevant to a particular planning role on a particular side. Planners can participate in setup, in step-wise extension of planning information, and in research gaming.

A significant feature of ConMod is that the generic military objects in the C^2 systems act as embedded autonomous planners within a military organizational hierarchy. Each C^2 object is capable of performing in an automatic or manual planning mode. The user can specify those objects that use automated planning or a level in an organization below which C^2 objects use automatic planning. Above the specified level, C^2 objects employ user provided plans and scripts for their planning. This insures that the simulation conforms to a concept of operation specified by the scenario.

Automated command and control refers to the capability of a C^2 object to adapt its plan to the developing situation. A C^2 object tracks its current situation (perceptions) and compares it with its desired situation (plan). Analogous with control systems, out of tolerance conditions generate corrective actions. A C^2 object can respond to a small deviation with a minor adjustment to a subordinate's directive. For major deviations which completely invalidate the existing plan, the entire planning process is reinitiated.

Each management object has both a deliberate planning cycle with a specified period and a current planning cycle for quick response. The emphasis in deliberate planning is to create favorable situations in the future. The emphasis in current planning is to counter unforeseen threats and to capitalize on unforeseen opportunities during execution of the deliberately prepared plan. Periodic and current planning cycles, combined with appropriate look ahead projections, allow for a mix of proactive and reactive behavior in automatic planners.

Object oriented implementation of a C^2 object conceals its actual planning methodologies from other simulation entities. Other entities interact with a planning entity only through received directives. This information hiding characteristic allows the use of any appropriate decision techniques within an object, from decision tables to complex artificial intelligence methods.

9.2 Automated Command and Control

In order to carry out automated command and control, three planning tasks have been identified. These are pathfinding, coordination, and strategy determination. They roughly correspond to tactical, operational, and strategic types of planning, although all methods may be used by units at any echelon.

The lowest level planning task is pathfinding. It locates a route between a starting point and an end point. The path is chosen based on a policy set which guides how the unit responds to such things as mobility, concealment, threat, etc.

The middle level planning task is coordination, or more precisely, cooperative, coordinating, parallel planning. This does such things as allocating resources among subordinates, resolving time constraints, etc. For example, it may control the use of a road net to get units to their objectives at the right times.

The highest level planning task is strategy determination. This consists of proposing a plan, then estimating the opponent's reaction, then reconsidering an alternate plan and estimating the new reaction, etc., for several iterations.

The system is hierarchical, corresponding to the military organizational hierarchy. A high-level unit is concerned mostly with large-scale strategy. It will work with a generalized and simplified representation of the battle area, emphasizing broad objectives, large-scale mobility corridors, and gross threats.

Plans generated at high levels are propagated downward. Middle-level planning units will emphasize the coordination of the disposition and movement of forces. Low-level planning units will emphasize tactical deployment details. However, each level may make some use of all three techniques.

Provision will be made for plan monitoring and repair. If events are proceeding within planning tolerances, no action is taken. If the bounds are exceeded, but still within the capabilities of the unit, the plan may be repaired. If outside its capabilities, the unit will pass the problem or opportunity up to higher authority.

These tasks will be implemented using techniques from various fields of artificial intelligence. The pathfinder will use the A* ("A-star") algorithm, which will be described presently. The coordinating planner will make parallel plans using expert system methodology, and it will coordinate tasks based on methods from robotics (for example, methods developed to coordinate the activities of two robot arms). The strategy planner will use methods from game theory, including alpha/beta pruning of min/max trees (a method for maximizing gains while minimizing losses). Plan repair and control will use augmented transition nets.

The pathfinder has been implemented, and implementation has begun on the strategy planner. The coordinating planner requires research effort in several areas, and work has commenced in one of these areas (Ref. 16).

The A* algorithm used in the pathfinder can be briefly described as follows. Basically, given start and end points, the algorithm seeks the path of minimum cost, where the cost depends on the policy set employed.

To do this, the algorithm starts by getting the next available path from a list of paths. If it is the goal path, it ends. Otherwise, it adds new paths to the list. The key is how to store and get the next paths. In our case, one efficient approach is based on what is called a priority queue (known as Dykstra's method). The priority is set by the cost (but it uses only costs determined in the past).

An even better approach is the A* algorithm. This bases priority on both the past cost and an estimate of the future cost. As an example, the A* cost function might use a set of coefficients for elevation, distance, concealment, mobility, and threat. The list can be expanded indefinitely. Each coefficient ranges from 0 to +1, where numbers near zero indicate a desire for avoidance. This coefficient set represents the policy set, which differs depending on each object. For example, an aircraft may use only minimum distance. An armored cavalry unit may want to avoid the enemy (minimizing the threat), or it may want to seek and destroy (maximizing the threat).

Constraints can be imposed to avoid certain areas of the terrain. This allows looking at alternate paths. That is, once the optimum path has been found, it is excluded, and the next search will yield the second best path. Hierarchical planning can also be done with the A* algorithm, with high-level planning on coarse, cognitive terrain, low-level planning on detailed, three-dimensional terrain.

10.0 ConMod Operational Applications

ConMod has the ability to represent the concepts of agility, initiative, depth, and synchronizaton that are important at the operational level of war.

Agility can be represented by decision cycle time characteristics. Since the relationships of the objects in the model correspond to the organization relationships in the force, and since the decision cycle times are represented for all objects in the force, the interrelationship of the command and control cycle times of the two sides can be represented.

Initiative is permitted in ConMod. The automated planning process involves look-ahead algorithms so that a proactive course of operation can be selected. That is, the model performs contingency analysis, including enemy intentions. The situation is monitored to allow reaction to unforeseen opportunities as well as unforeseen problems. An explicit organizational hierarchy allows tactical success to leverage initiative at higher operational levels.

Depth is modeled through the explicit representation of the organization hierarchy. This allows each level to respond to the situation it perceives in accordance with its characteristic time, space, and resource constraints. The model can have several agents at any organizational level. Some can be responsible for the deep battle, others for the rear battle, others for the close battle. Each has separate resources and separate planning processes.

Synchronization is dealt with by having many directives derive from one plan. Since the planning agents are explicity planning for dissimilar subordinates, the dissimilar subordinates will behave in accordance with the single plan of their superior. Hence synchronization of effort occurs at every organizational level.

ConMod has been developed in a militarily balanced environment, i.e., with both Army and Air Force members on the project team. This has facilitated joint input to the functional representations for air/land battle. This joint development has also encouraged equivalent and balanced representation of joint operational elements.

Because the ConMod structure does not dictate any particular organizational level, echelon-above-corps features can be represented as natural extensions of organizational functions to higher levels.

11.0 Input and Output Data

In this section, we will address the data aspects of ConMod, both what it requires as input and what it generates as output. We will first relate the data handling phases of the analysis cycle to the overall requirements for studies as well as training, etc. We will then describe the two main classes of data required by ConMod, that related to command and control and that related to physics processes. We will then discuss the output produced by ConMod and methods for handling it.

The need for minimizing cycle time ***from question asked to answer provided*** leads us to consider a user-oriented system. Running the simulation is only a small part of analyzing an issue. What counts is the entire cycle from the time a decision maker asks a question to the time the decision maker receives the answer. Therefore, although run time is important in the design of the ConMod simulation software, methods of shortening the time to handle input data and analyze output data are also taken into consideration in the overall ConMod project.

To make the best use of limited analyst resources, the input data for the model must be easy to set up and modify, and the output must be capable of rapid analysis and presentation. This implies the use of interactive color graphics systems to provide user-friendly interfaces.

11.1 Input Data Requirements

For ease of setup, the ability exists to manage multi-study libraries through a database, as well as the ability to create new scenarios as derivatives of previous studies. ConMod also makes use of preprocessing utilities to create the terrain and force organization parts of the scenario. These utilities use INGRES, which is the Army standard database.

A major use of ConMod is as an analytic tool to evaluate system effectiveness. Thus effectiveness must appear as an output, not an input. The system input is therefore restricted to engineering data, as distinguished from effectiveness data. This includes characteristic parameters for moving, sensing, engaging, and communicating.

ConMod uses measurable data from responsible laboratories to the maximum extent possible. It provides assessment audit trail to laboratory data. This minimizes aggregation methodology validation and consistency checking by restricting aggregation to match laboratory data. It allows for representing individual high value systems.

In addition to data for weapons systems, ConMod requires data on military organizations. This includes not only their structure but also their policies, etc., which govern the way they act.

ConMod's control system approach permits a close relationship to real-world decision processes. On the physics side, entities are modeled using engineering data, which is readily measured. Similarly, on the cognitive side, the command and control entities are intended to correspond closely with those in the real world. This allows decision data to be obtained as simply as possible.

For example, a cognitive entity may be a company. According to current doctrine, a U.S. Army company commander uses the METT-T acronym to remind himself to take mission, enemy, terrain, troops, and time available into account when making his plans. This maps into the generic management model used for command and control entities in ConMod, which allows directly incorporating this particular doctrinal approach in the model.

To enter the planning information, we provide a family of functionally specific, interactive graphic, planning utilities. Examples are the maneuver planner's planning utility, fire support planning utility, etc.

Because of the automated planning capability, automated planners can assist in positioning simulated units during the setup mode.

11.2 Output Data

The quantities of data and calculations seem large, but they may be less than the combination of high resolution and large scope implies at first glance. While there will be many items in the model, they consist of many replications of a few types. The characteristic data for a type may be extensive, but it only has to be stored once.

Similarly, physics processes for engagements require a great deal of computation, but only a fraction of the units in a corps are engaged at any one time. The process tends to be self limiting. If the battle is intensive, then attrition is high, and the number of units decreases rapidly.

In order to make use of this information for tracking cause and effect, it is necessary to record it. ConMod will have the capability of recording every event, whether physical or cognitive. While this has the advantage of making results available for future studies, it leads to a vast quantity of output. To access this through a relational database may require substantially more time than to redo the ConMod calculation. Consequently, ConMod is planned for the archival of a set of specific events or types of events, to be requested by the analyst.

Physical processes require high resolution, while cognitive processes use aggregated perceptions. The physics model in ConMod uses ground truth. Engaged systems will be represented in high resolution on 3-D terrain. While this information is available for analysis, it need not be presented unless required. Instead, what will be displayed is a perception expressed in terms of aggregated units.

ConMod provides a selective view. The aggregation in the model is linked to the data, but the aggregation that the user views and considers is linked to the problem under study. Since the organizational hierarchy is represented, the user can examine the conflict at the level of interest within the model organization. The advantage of this is that issues identified in the analysis at the level of interest can be resolved if desired by examination of additional, underlying detail.

For example, two Blue tanks might be engaging three Red tanks. This is ground truth. The perceived situation may be that a Blue platoon is engaging a Red company. The number of tanks probably disagrees with the usual numbers in these units. On a tactical view, the two tactical groupings, platoon and company, will be shown by their symbols, and engagement will be indicated.

These perceptions may be aggregated to any desired level. If it is a corps level simulation, the corps commander may want to issue directives only one echelon down, to divisions, and he may want to see a display extending only two levels down, to brigades. However, just as in the real battle, individual tanks are in action, and the results of individual engagements determine the course of events.

Post-run analysis is supported through graphics queries of the history database. Graphics queries can be in time and space, by activity, by who did it, etc.

12.0 ConMod Computer Resource Estimates

The computer resources needed for the ConMod prototype were estimated for three areas: CPU time, central memory, and mass storage.

The analysis was based on the ConMod prototype baseline scenario. This is a corps level engagement in size and intensity, with seven brigades in close combat at any one time. The resource estimates were made for seven days of battle. Planning event computational cost estimates were included. For timing calibrations, Janus regiment vs battalion size simulations were used. The estimates are for the VAX-8800 development machine.

The difficulty here is in trying to develop estimates for a software package before it has been built. Fortunately, a somewhat similar model is available on the same machine, namely, Janus. Janus contains the same type of computational algorithms that ConMod will have and thus can be used for calibration. For this purpose a calibration scenario was constructed to run in non-interactive mode.

In the calibration scenario, a Red regiment is attacking a Blue battalion. They engage in all-out, heavy combat. Three-dimensional, digitized terrain was used, located in Korea. The scenario included direct fire, artillery, smoke, and engineered barriers, with movement, line-of-sight acquisition, and event reports. The outcome was a drawdown of 70% of both forces. Simulating 30 minutes of battle took less than 3.5 minutes of CPU time on the VAX-8800.

For the ConMod baseline scenario, the battle plan implies that 21 Blue battalions fight 17 Red regiments. This is equivalent to 21 Janus calibration scenarios, each of which may be regarded as a Major Engagement. In terms of the baseline scenario, a Major Battle occurs when 21 Major Engagements occur simultaneously.

Over a 7-day period, the baseline scenario assumes that Major Battles occur according to the following schedule:

Days	Major Battles per day	Major Engagements per day
1	3	63
2-4	2	42
5-7	1	21

A total of 252 Major Engagements occur, and 882 CPU minutes are required to simulate seven days of combat on the VAX-8800.

Regarding the ConMod automated planning process, estimates were made that indicated an average planning event in ConMod will take 100 times as long as an acquisition type computational algorithms in Janus. The average rate of planning events for the ConMod prototype scenario is 30 per minute. The event timings from the Janus calibration scenario were used to estimate that 458 minutes of CPU time will be needed for the automated planning of the seven days of simulated combat.

The results of this calibration indicate that the simulation of 7 days of combat requires 882 CPU minutes for combat plus 458 minutes for automated planning, or a total of 1340 CPU minutes. Seven days is 10080 minutes. Therefore, the simulator will run in the ratio of 10080/1340, or 7.5 times faster than real time on a single processor.

For central memory requirements, it was estimated that 30 megabytes are needed for code and data for the seven days of combat. The VAX-8800 currently has 48 megabytes, so there is a significant margin.

For mass storage requirements, no problem was anticipated in storing the results on disks.

Some general observations may be made here. The VAX-8800 has two identical processors that share central memory. The utilization of these processors is controlled by the operating system. A user with achieve maximum benefit from the second processor by executing two independent jobs (simulations) simultaneously. In this event the effective throughput rate is doubled.

The timing analysis is very conservative in many respects. Attrition is expected to considerably reduce run times. Janus and other similar simulators were designed to operate on older-technology machines, which had small central memories. ConMod algorithms will be designed to utilize the larger central memory to provide faster execution.

In summary, a conservative analysis predicts that ConMod operating on a VAX-8800 will run substantially faster than real time in a simulation mode. Central memory and mass storage requirements will impose no limitations.

ACKNOWLEDGEMENTS

This paper reports the contributions of numerous individuals on the Conflict Model project team and in the Conflict Simulation Center during the period up to July 1987. Yu-Hsing Chiu was the ConMod project leader. The System Environment team was led by Sara M. Strange and included John S. Hazard and Norman J. Cushing. The Simulation Model team was led by Robert L. Pancotti and included John R. Tannahill, Stephen Wong, Mitch M. Lee, and D. Lynn Hughes. Stephen Wong also headed the QA effort. The Military System team, led by Charles D. Jones, included Gregory D. Kreis, Thomas F. Hendrickson (until April 1987), Ralph M. Toms, and Jeffrey A. Kueffer, with contributions by Michael J. Uzelac and Jeffrey E. Pimper. Ronald L. Burback, aided by David C. O'Keefe, developed the automated command and control area. S. Christian Simonson was the Conflict Simulation Center project manager.

References

1. Memorandum of Agreement between the Department of the Army and the Department of Energy and the Lawrence Livermore National Laboratory, August 15, 1985.

2. *Revised Requirements Document for the Army / Lawrence Livermore National Laboratory Corps Model*, Army Model Improvement Program Management Office, Ft. Leavenworth, KS, 6 December 1985.

3. *LLNL Conflict Model Concept Document*, Conflict Simulation Center, Lawrence Livermore National Laboratory, Livermore, CA, 11 March 1986.

4. *Software Configuration Management Plan for the LLNL Conflict Model LCM) Project*, Conflict Simulation Center, Lawrence Livermore National Laboratory, Livermore, CA, 18 June 1986.

5. *Software Quality Assurance Plan for the LLNL Conflict Model (LCM) Project*, Conflict Simulation Center, Lawrence Livermore National Laboratory, Livermore, CA, 1 July 1986.

6. *The Conflict Model (ConMod) Requirements Documentation*, Vol. 1, "Requirements," Conflict Simulation Center, Lawrence Livermore National Laboratory, Livermore, CA, November 24, 1986.

7. *The Conflict Model (ConMod) Requirements Documentation*, Vol. 2, "Data/Control Flow Diagrams," Conflict Simulation Center, Lawrence Livermore National Laboratory, Livermore, CA, November 24, 1986.

8. *The Conflict Model (ConMod) Requirements Documentation*, Vol. 3, "Requirements Dictionary," Conflict Simulation Center, Lawrence Livermore National Laboratory, Livermore, CA, November 24, 1986.

9. *The Conflict Model (ConMod) Design Documentation*, Vol. 1, "Simulation Overview," Conflict Simulation Center, Lawrence Livermore National Laboratory, Livermore, CA, July 6, 1987.

10. *The Conflict Model (ConMod) Design Documentation*, Vol. 2, "Simulation Program Units," Conflict Simulation Center, Lawrence Livermore National Laboratory, Livermore, CA, July 6, 1987.

11. *The Conflict Model (ConMod) Design Documentation*, Vol. 3, "System

Support Environment," Conflict Simulation Center, Lawrence Livermore National Laboratory, Livermore, CA, July 6, 1987.

12. Letter to BG John D. Robinson, Commander TRADOC Analysis Command, from Donald K. Blumenthal, LLNL, April 22, 1987.

13. Y.-H. Chiu, C. D. Jones, G. D. Kreis, S. C. Simonson, S. M. Strange, J. R. Tannahill, "The Simulation of Command and Control in the Conflict Model (ConMod)," *Proceedings of the Joint Directors of Laboratories 1987 Command and Control Research Symposium,* National Defense University, Washington, DC, UCRL-97281 Preprint (1987).

14. *INGRES Reference Manual*, Relational Technology, Inc., Alameda, CA (1986).

15. J. R. Tannahill, S. C. Simonson, *Choice of the Ada® Programming Language for the Conflict Model (ConMod),* Lawrence Livermore National Laboratory, Livermore, CA (1987, in press).

16. R. L. Burback, "PROVER: a First-Order Logic System in Ada", 3rd Annual Conference on Artificial Intelligence and Ada, Fairfax, VA, UCRL-96632 Preprint (1987).

17. *The Janus Manual*, Version 3.02, Conflict Simulation Center, Lawrence Livermore National Laboratory, Livermore, CA (1986).

Appendix A

MEMORANDUM OF AGREEMENT
BETWEEN
THE DEPARTMENT OF THE ARMY
AND
THE DEPARTMENT OF ENERGY
AND
THE LAWRENCE LIVERMORE NATIONAL LABORATORY

1. INTRODUCTION. The Lawrence Livermore National Laboratory (LLNL) under contract through the University of California with the Department of Energy (DOE), the Training and Doctrine Command (TRADOC) of the Department of the Army (The Army), and the Department of the Air Force have been engaged in cooperation regarding computer combat simulation models. LLNL possesses a unique capability applicable to the development of certain computer models of interest to the Army and is a DOE Federally Funded Research and Development Center (FFRDC). TRADOC and LLNL have concluded an agreement of 6 February 1981 concerning establishment and staffing of a TRADOC Element Lawrence Livermore (TELL). LLNL is currently beginning the development of a new Joint AirLand Corps level computer combat model. The Army also has a requirement for a new Corps level model. The Army wishes to be a co-sponsor with LLNL in a cooperative program to produce a Joint AirLand Corps level combat model.

2. PURPOSE. This Memorandum of Agreement (MOA) sets forth the general principles and procedures whereby the Army will co-sponsor with LLNL the development of a Joint AirLand Corps level computer combat simulation software system and its associated documentation. Both parties understand that this agreement will result in the following primary products:

a. A jointly approved model requirements document.

b. An automated AirLand Combat Model developed to meet the agreed upon requirements.

3. SCOPE OF PROGRAM. LLNL will develop a high resolution Joint AirLand Corps level computer combat simulation model which will incorporate the jointly approved requirements of LLNL and the Army. The envisioned model characteristics include: AirLand Corps level, automated, and interactive. The principal Army emphasis is on the development of an automated model which must include automated command and control functions. The development will address minimizing the total response cycle of the model to include data input and formulation, scenario development, model run speed, and post-processor/analysis of model output. The model will be used for

analysis, research gaming, and training. LLNL and the Army will support this model development with agreed level of funding, personnel resources, and facilities. The development effort is anticipated to cover approximately three years.

4. LLNL UNDERTAKINGS. DOE agrees that LLNL will:

a. Participate through the technical steering group in the development and review of model requirements and the review of model development.

b. Design and develop a Joint AirLand Corps level computer combat model with associated documentation. The model will meet Army and LLNL requirements that have been approved by the co-chaired technical steering group.

c. Prepare a model development plan to be approved by the technical steering group.

d. Provide reports to the Army keyed to critical decision points as per the model development plan.

e. Provide the facility and computer support necessary for this model development.

f. Provide the management and staff for the development of this model.

g. Designate an individual to serve as co-chairperson of the technical steering group.

h. Upon completion of the development, provide the Army with complete model documentation to include a copy of the model in machine readable form, object code, listings of the source code, and users manuals.

5. ARMY UNDERTAKING. The Army will:

a. Participate through the technical steering group in the development and review of model requirements and the review of model development.

b. Designate the Technical Advisor to the Deputy Chief of Staff for Operations and Plans of the Army as the co-chairperson of the technical steering group.

c. Provide a funding level of approximately $1,000,000. per year for three years to support development of the Joint AirLand Corps level model.

d. Designate the Director, Army Model Improvement Program Management Office (AMMO), as the single Army point of contact to coordinate technical, resource, and management support required from the Army under this agreement.

e. Continue to staff the TELL with three officers pursuant to the terms of the Memorandum of Agreement of February 6, 1981. The Army affirms its commitment to the qualification requirements of Paragraph 4 of the Memorandum of Agreement of February 6, 1981.

6. TECHNICAL STEERING GROUP. A technical steering group will be established and chaired jointly by LLNL and the Army. The initial task of the co-chairmen will be to establish procedures for the development, review, and approval of model requirements. Once model requirements are approved, LLNL will begin formal design and development of the model. Changes to approved requirements will incur time and resource penalties and must be mutually agreed to by the Army and LLNL. Periodic reviews of the program will be conducted by the group at the call of either co-chairperson, but not less than twice a year.

7. ADMINISTRATION.

a. LLNL effort will be performed pursuant to contract W-7405-ENG-48 between the U.S. Department of Energy and the University of California. This Memorandum provides Agreement in Principle as to the relationship between the Army and LLNL and in no way supersedes the applicable contract which is legally controlling.

b. The parties recognize that implementation of the provisions of this Memorandum of Agreement is subject to the availability of funds and compliance with existing statutes and federal acqusition regulations. LLNL will not initiate any work in support of Army requirements until execution of the necessary contractual documentation by the San Francisco Operations Office of the Department of Energy. The Army will place orders under the provisions of 31 U.S.C. 1535 with the Department of Energy San Francisco Operations Office for work to be performed by LLNL pursuant to this agreement. The purpose of such orders is to capitalize on the unique capabiliities of LLNL, and maintain an essential engineering, research and development capability at LLNL, a DOE-sponsored FFRDC, and is not intended to circumvent the requirements of 41 U.S.C. 253(f)(5)(B) or 10 U.S.C. 2403(f)(5)(B).

c. Any guidance or direction which modifies the scope of work to be performed by LLNL shall be coordinated with the DOE San Francisco Operations Office. Technical exchange consistent with the DOE approved scope of work will be directly between the Army and LLNL.

d. The cost and scope of specific work to be performed by LLNL will be established pursuant to normal contract procedures and incorporated in applicable contract documentation.

8. DURATION AND REVIEW. The duration of this Memorandum of Agreement is for three years from the above date of last signature. The Parties may terminate this Memorandum of Agreement at any time upon notice to the other signatories. Termination of implementing contracts will be executed pursuant to the terms of such contracts. Termination of this Memorandum of Agreement by LLNL or the University of California will have no effect upon the obligations of these entities under contract with Government then existing.

9. This Memorandum of Agreement supersedes any inconsistent provisions of the Agreements cited in Paragraph 1 above.

Walter W. Hollis	Kathleen M. Day	W. B. Shuler
US Army	San Francisco Operations Office, DOE	LLNL
Date 8/1/85	Date 8/15/85	Date 8/2/85